

Universidad ORT Uruguay
Facultad de Ingeniería



Diseño de Aplicaciones 2

Obligatorio 1

Martin Civetta (271063)
Ignacio Etcheverry (279855)

<https://github.com/NACHO9999/OB-DAP2>

2 de Mayo del 2024

Índice

Índice.....	2
Git Flow.....	3
Tabla API.....	4
Diagrama de Paquetes	9
UML.....	10
Funcionamiento.....	11

Git Flow

Nuestra Adaptación de Git Flow

En el desarrollo de este proyecto, decidimos implementar la práctica estándar de Git Flow, con el objetivo de mantener una estructura clara y ordenada que facilitara la colaboración y la gestión del código. A continuación, detallo cómo organizamos este flujo:

-> **Main:** Esta es nuestra línea base. Representa la versión más estable de nuestro proyecto y solo se actualiza cuando estamos listos para lanzar una nueva versión oficial. Durante todo el proceso de desarrollo, los únicos commits que se implementaron en esta rama fueron los que estábamos seguros que funcionaban para ya tener una base con que poder seguir el flujo de trabajo tranquilamente.

-> **Develop:** Es acá donde sucede la mayoría de nuestra actividad de desarrollo. Todas las contribuciones, una vez revisadas y aprobadas, se incorporan en esta rama. Además, siempre que empezamos a trabajar en una nueva característica o corrección, usamos "Develop" como punto de partida.

Tabla API

API	DESCRIPTION	REQUEST BODY	RESPONSE BODY
POST /api/administradores	Crear un nuevo administrador	{ "nombre": "Juan", "apellido": "Perez", "email": "juan.perez@example.com", "contraseña": "segura123" }	{ "status": 201, "data": { "id": 1, "email": "juan.perez@example.com" } }
POST /api/invitaciones	Invitar a un nuevo usuario a la plataforma	{ "email": "ejemplo@gmail.com", "nombre": "Ana", "fechaLimite": "2024-05-30" }	{ "status": 201, "data": { "id": 1, "email": "ejemplo@gmail.com" } }
DELETE /api/invitaciones/{email}	Eliminar una invitación no aceptada	Ninguno	{ "status": 200 }
POST /api/categorias	Dar de alta nuevas categorías para las solicitudes	{ "nombre": "Electricista" }	{ "status": 201, "data": { "id": 1, "nombre": "Electricista" } }
POST /api/edificios	Crear un nuevo edificio	{ "nombre": "Edificio Central", "direccion": "123 Calle Principal", "ubicacion": "Centro Ciudad", "empresaConstructora":	{ "status": 201, "data": { "id": 1, "nombre": "Edificio Central" } }

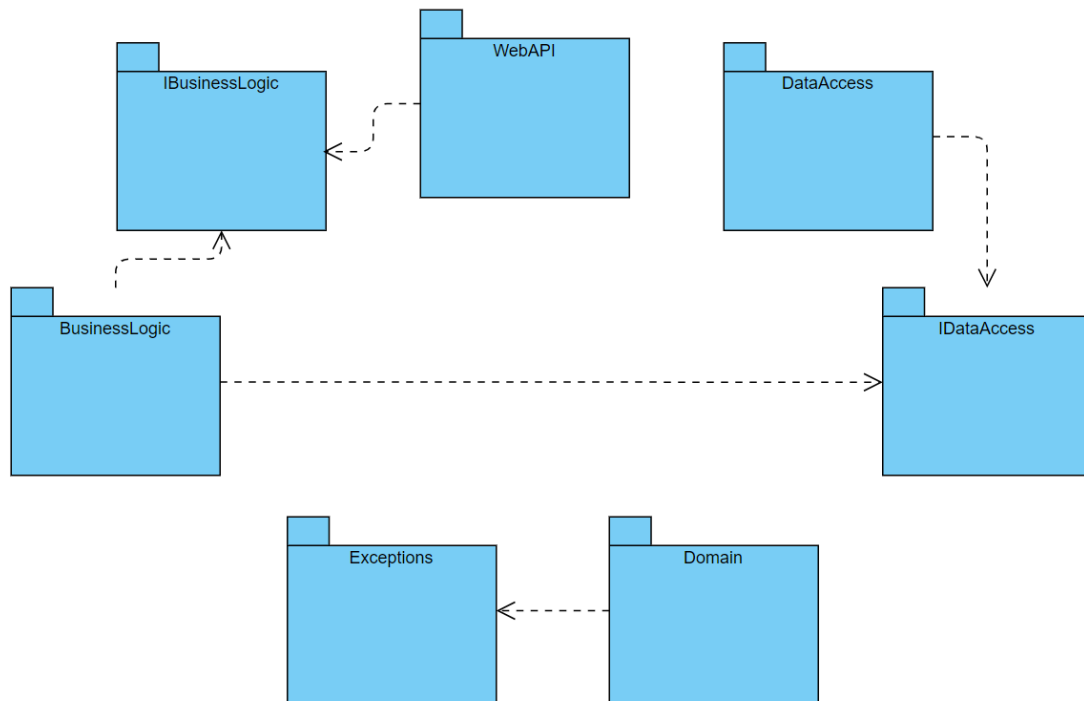
		"Construcciones ABC", "gastosComunes": 1200, "departamentos": [{ "piso": 1, "numero": 101, "dueño": {"nombre": "Ana", "apellido": "Ruiz", "email": "ana.ruiz@example.com "}, "cantidadCuartos": 3, "cantidadBaños": 2, "terrazza": true }] }	
PUT /api/edificios/{id}	Actualizar detalles de un edificio	{ "nombre": "Edificio Central Mod", "direccion": "123 Calle Principal", "ubicacion": "Centro Ciudad", "empresaConstructora": "Construcciones ABC Mod", "gastosComunes": 1300 }	{ "status": 200, "data": { "id": 1, "nombre": "Edificio Central Mod" } }
DELETE /api/edificios/{id}	Eliminar un edificio	Ninguno	{ "status": 200 }
POST /api/usuarios/mantenimiento	Crear un usuario de mantenimiento	{ "nombre": "Carlos", "apellido": "Constructor", "email": "carlos.constructor@exa mple.com", "contraseña":	{ "status": 201, "data": { "id": 1, "email": "carlos.constructor@exa mple.com" } }

		"construir123" }	
GET /api/solicitudes	Obtener todas las solicitudes, filtrable por categoría	Ninguno	{ "status": 200, "data": [{"id": 1, "descripcion": "Fuga de agua", "estado": "abierto", "categoria": "Fontanero"}] }
POST /api/solicitudes	Crear una solicitud de servicio	{ "descripcion": "Fuga de agua", "departamentoid": 1, "categoriald": 2 }	{ "status": 201, "data": { "id": 1, "estado": "abierto" } }
PUT /api/solicitudes/{id}/asignar	Asignar una solicitud a una persona de mantenimiento	{ "usuarioid": 2 }	{ "status": 200 }
PUT /api/solicitudes/{id}/atender	Marcar una solicitud como 'en atención'	Ninguno	{ "status": 200, "data": { "id": 1, "estado": "atendiendo", "fechaInicio": "2024-04-01T10:00:00" } }
PUT /api/solicitudes/{id}/completar	Marcar una solicitud como 'completada'	{ "costoTotal": 150 }	{ "status": 200, "data": { "id": 1, "estado": "cerrado", "fechaFin": "2024-04-02T17:00:00" } }
GET /api/reportes/solicitudes/edificio/{edificio}	Ver solicitudes agrupadas por edificio	Ninguno	{ "status": 200, "data": [{ "edificio": "Edificio Central",

old}			"solicitudesAbiertas": 4, "solicitudesCerradas": 2, "solicitudesAtendiendo": 10 }} }
GET /api/reportes/solicitudes/mantenimiento	Ver solicitudes por persona de mantenimiento	{ "nombre": "Carlos" }	{ "status": 200, "data": [{"nombreMantenimiento": "Carlos", "solicitudesAbiertas": 3, "solicitudesCerradas": 9, "solicitudesAtendiendo": 0, "tiempoPromedioCierre": "5hs"}] }
GET /api/reportes/solicitudes/categoria	Ver solicitudes por tipo (categoría)	{ "edificioId": 2 }	{ "status": 200, "data": [{"categoria": "Electricista", "solicitudesAbiertas": 2, "solicitudesCerradas": 9, "solicitudesAtendiendo": 0}, {"categoria": "Vecino molesto", "solicitudesAbiertas": 1, "solicitudesCerradas": 10, "solicitudesAtendiendo": 0}] }
POST /api/usuarios/encargados	Alta de usuarios encargados	{ "nombre": "Laura", "apellido": "Mendez", "email": "laura.mendez@example.com", "contraseña":	{ "status": 201, "data": { "id": 1, "email": "laura.mendez@example.com" } }

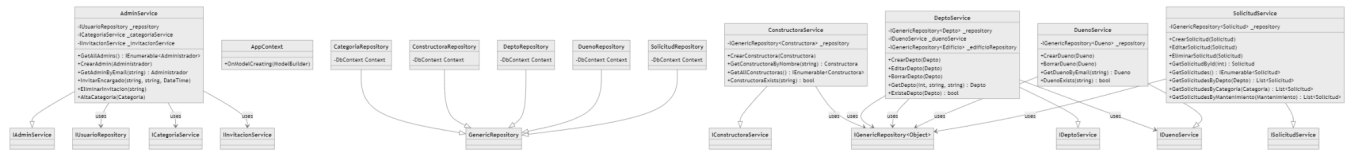
		"secure456" }	
DELETE /api/usuarios/encargados/{id}	Eliminar usuarios encargados	Ninguno	{ "status": 200 }
PUT /api/invitaciones/{id}/modificar	Modificar invitación para extender la fecha límite	{ "nuevaFechaLimite": "2024-06-01" }	{ "status": 200, "data": { "id": 1, "fechaLimite": "2024-06-01" } }

Diagrama de Paquetes



En este obligatorio decidimos separar nuestro proyecto en partes distintas con responsabilidades definidas. Seguimos los principios solid para separar las dependencias y que de ser necesario, los paquetes dependen de abstracciones de módulos menores y no de los módulos mismos. Seguimos los principios de clean code basándonos en la nomenclatura de C#.

Diagrama de Clases (UML)



Para verlo mejor en:

<https://drive.google.com/drive/folders/14suBsMOzCI8LzVBW4yTHknuQcZ7K4OS5?usp=sharing>

Funcionamiento del Programa

ob-backend

- Exceptions
- ob.BusinessLogic
- ob.DataAccess
- ob.Domain
- ob.IBusinessLogic
- ob.IDataAccess
- ob.WebApi

ob.BusinessLogic

- bin
- obj
- AdminService.cs
- CategoriaService.cs
- ConstructoraService.cs
- DeptoService.cs
- DuenoService.cs
- EdificioService.cs
- EncargadoService.cs
- InvitacionService.cs
- MantenimientoService.cs
- ob.BusinessLogic.csproj
- SessionService.cs
- SolicitudService.cs

ob.DataAccess

- bin
- Migrations
- obj
- AppContext.cs
- appsettings.json
- CategoriaRepository.cs
- ConstructoraRepository.cs
- DeptoRepository.cs
- DuenoRepository.cs
- EdificioRepository.cs
- GenericRepository.cs
- InvitacionRepository.cs
- ob.DataAccess.csproj
- SessionRepository.cs
- SolicitudRepository.cs
- UsuarioRepository.cs

ob.Domain

- bin
- Enums
- obj
- Administrador.cs
- Categoria.cs
- Constructora.cs
- Depto.cs
- Dueno.cs
- Edificio.cs
- Encargado.cs
- Invitacion.cs
- Mantenimiento.cs
- ob.Domain.csproj
- Session.cs
- Solicitud.cs
- Usuario.cs
- Validator.cs

ob.WebApi

- bin
- Controllers
- DTOs
- Filters
- obj
- Properties
- appsettings.Development.json
- appsettings.json
- ob.WebApi.csproj
- ob.WebApi.http

11